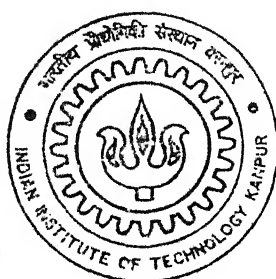# MULTI-OBJECTIVE OPTIMIZATION OF SEMI-BATCH COPOLYMERIZATION REACTORS USING ADAPTATIONS OF GENETIC ALGORITHM (GA)

By

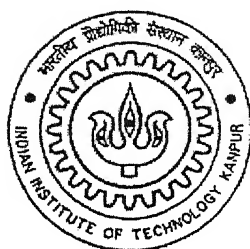## Arpan Nayak

DEPARTMENT OF CHEMICAL ENGINEERING

Indian Institute of Technology Kanpur

FEBRUARY, 2003

MULTI-OBJECTIVE OPTIMIZATION OF SEMI-BATCH COPOLYMERIZATION REACTORS USING ADAPTATIONS OF GENETIC ALGORITHM (GA)

by

Arpan Nayak

DEPARTMENT OF CHEMICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

February, 2003
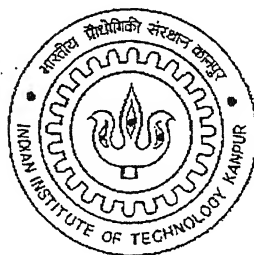
143448

MULTI-OBJECTIVE OPTIMIZATION OF SEMI-BATCH COPOLYMERIZATION REACTORS USING ADAPTATIONS OF GENETIC ALGORITHM (GA)

A Thesis Submitted
in Partial Fulfillment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

by

ARPAN NAYAK

to the

DEPARTMENT OF CHEMICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

February, 2003

"Shraddhaanjali"

……….. to the loving memories of my
grandparents

who built the pedestal of my education

CERTIFICATE

This is to certify that the work contained in this thesis entitled 'MULTI-OBJECTIVE OPTIMIZATION OF SEMI-BATCH COPOLYMERIZATION REACTORS USING ADAPTATIONS OF GENETIC ALGORITHM (GA)'has been carried out by **Arpan Nayak** under my supervision and that this work has not been submitted elsewhere for a degree.

February, 2003

Dr. Santosh K. Gupta

Professor
Department of Chemical Engineering
Indian Institute of Technology, Kanpur
India.

Acknowledgement

At the very beginning I would like to thank the authority of IIT, Kanpur for providing high-class facilities and the ambience necessary to carry out research work.

During this research work, my thesis guide **Prof. S. K. Gupta** has helped me immensely with his ingenious ideas and valuable guidance. I wish to express my profound sense of gratitude and sincere regards to him for motivating and encouraging me during all the stages of my thesis work. By working under his aegis I have come to know the intricate details of real research. His guidance will act as a beacon of light throughout my life.

I would like to thank Rahul Kasat for providing me codes of Genetic Algorithm in FORTRAN.

I shall always cherish the sweet memories attached with Ami, Anjana, Chandanda, Jitendra, Patil, Sameer, Shankar, Shrikant, Sunil, Swarnendu and Tejas.

The time spent with Tirthada, Kanuda, Jyotida, Sauravda, Parthada, Prakashda, Dulalda, Sayan, Braja, Giri, Minu, Deba, Dam, Asit, Arindam, Samui, AP, Liton, Chat and Suman in G-top and H-top has been a major recreation for me. The memories associated with them will surely be cherished forever.

I would like to thank Shubhra aunty for her motherly love and affection. The delicious dishes prepared by her were of special attraction during occasional visits to Sir's residence.

This acknowledgement would remain incomplete if I fail to remember my **parents** and all other **family members**. Their silent presence behind all my achievements cannot be properly expressed.

Finally, I would like to thank DST for providing the financial support during the completion of this work.

February, 2003   Arpan Nayak

Contents

List of Figures

List of Tables

Nomenclature

F_i	Mole fraction of ith monomer in copolymer; $i = 1, 2$
F	Fitness function
f	Initiator efficiency
I	Initiator concentration in reaction mixture (mol l^{-1})
I_i	i^{th} objective function; $i = 1, 2, 3$
I_f	Initiator concentration in feed (mol l^{-1})
k_d	Initiator decomposition rate constant (s^{-1})
k_{fij}	Chain transfer rate constant; $i, j = 1, 2$ (l mol^{-1} s^{-1})
k_{pij}	Propagation rate constant; $i, j = 1, 2$ (l mol^{-1} s^{-1})
k_{tcij}	Combination termination rate constant; $i, j = 1, 2$ (l mol^{-1} s^{-1})
k_{tdij}	Disproportionation termination rate constant; $i, j = 1, 2$ (l mol^{-1} s^{-1})
l_{str}	No. of binaries per substring
M_i	i^{th} monomer concentration in reaction mixture; $i, j = 1, 2$ (mol l^{-1})
M_{if}	i^{th} monomer concentration in feed; $i, j = 1, 2$ (mol l^{-1})
M_n	Total number average molecular weight
$M_{n,s}$	Desired value of total number average molecular weight
M_w	Total weight average molecular weight
N_{gen}	Maximum number of generations
N_{ga}	Number of discrete points generated in the range 0 to t_f for each decision variable
N_{seed}	Random seed number

N_p	Population size
P_i	i^{th} moment of the total number molecular weight distribution of radicals of type-1; $i = 0, 1, 2$
p_c	Probability of crossover
p_m	Probability of mutation
$P_{n,m}$	Concentration of growing polymer containing n units of monomer-1 and m units of monomer-2 and ending in monomer-1 unit (mol l^{-1})
Q	Polydispersity index
Q_i	i^{th} moment of the total number molecular weight distribution of radicals of type-2; $i = 0, 1, 2$
$Q_{n,m}$	Concentration of growing polymer containing n units of monomer-1 and m units of monomer-2 and ending in monomer-2 unit (mol l^{-1})
r_i	Reactivity ratio; $i = 1, 2$
T	Reactor temperature (K)
TOL_M	Tolerance on M_n constraint violation
TOL_F	Tolerance on F_1 constraint violation
t	Time (min)
t_f	Final time of reaction (min)
u	Continuous feed rate (l min^{-1})
$\mathbf{u}$	Vector of control variables
V	Reactor volume (l)
w_i	Molecular weight of ith monomer; $i = 1, 2$ (g mol^{-1})
x_i	Fractional conversion of ith monomer at any time; $i = 1, 2$

x   Vector of state variables

x_0 Initial value of the state vector

Greek symbols

λ_k k^{th} moment of the total number molecular weight distribution of the dead

             copolymer; $k = 0, 1, 2$

ϕ Molar ratio of unreacted monomers in reaction mixture

ϕ_f Molar ratio of monomers in feed

ϕ_s Desired value of molar ratio of unreacted monomers in reaction mixture

ϕ_t Cross termination factor

Subscripts

0   Initial condition

f   Feed condition

s   Desired value

ABSTRACT

The polymerization of styrene-acrylonitrile (SAN) random copolymer in semi-batch reactors is optimized using several objective functions that are often conflicting and non-commensurate in nature. These include the average copolymer composition, its number-average molecular weight, its polydispersity index and the conversion attained in the reactor. Two decision/control variables are used, namely, the rate of addition of a continuous monomer-solvent-initiator mixture (constant composition), and the reactor temperature. The elitist non-dominated sorting genetic algorithm is adapted for decision variables that are *functions* of time (trajectory optimization), and used. This robust, AI (artificial intelligence)-based technique, enables the solution of far more complex optimization problems than reported in the literature. A set of several non-dominating (equally good) Pareto optimal solutions is obtained. These provide insights into the conflicting nature of the objective functions involved. An engineer (decision maker) can use his judgment (often intuitive) to choose the preferred solution from among these possibilities.

The design and operation of industrial polymerization processes is intimately related to the accurate control of the operating conditions that are, in turn, related to the properties of polymers. It is important to (i) compute the optimal operating conditions (off-line optimization) required to produce polymer products having specified properties, and then to (ii) implement this on the industrial unit, making sure that the effect of disturbances is negated (on-line control). The latter may pose severe problems, partly because of the lack of accurate on-line sensors[1] for the measurement of the properties of the reaction mass, such as the average molecular weight (or its distribution, the MWD), and the monomer conversion. The focus of the present study is on the first of these two stages. This step is important if we wish to minimize the production of off-spec (off-specification) products, while simultaneously maximizing the productivity or profitability of the unit.

Although continuous processes produce polymers having more consistent quality in large volumes (through process automation), batch or semi-batch reactors are often used for producing small- to medium-volume polymers. Indeed, several specialty polymers, including copolymers, are produced in batch or semi-batch reactors. In this work, we focus on the off-line optimization of the production of copolymers in semi-batch reactors.

The optimization of free-radical copolymerization reactors is far more complex than that of homo-polymerizations, since controlling the number average molecular weight (and/or the MWD through the polydispersity index, Q) and the monomer

conversion is not sufficient. For these systems, obtaining polymer having a desired value of the average copolymer composition is equally important. It is well known[2] that two styrene-acrylonitrile (SAN) copolymers differing by more than 4% in the average acrylonitrile content are incompatible, resulting in poor physical and mechanical final properties. Due to the different reactivities of the comonomers in free-radical copolymerization processes, composition drift occurs unless a mixture having higher concentrations of the more reactive monomer is added continuously (semi-batch mode) to the reactor. One, therefore, needs to determine how this mixture should be added to the reactor, as well as how the temperature in the reactor should be varied with time, so as to produce polymer having the desired properties, optimally (say, in as short a reaction time as possible). These are the two commonly used control or decision variables. The objective functions and constraints could be selected from among the following:

 (a) obtain polymer having the desired value of the average composition,

 (b) obtain polymer having the desired value of the number-average molecular weight (or, minimize the deviations of these two properties from their desired values),

 (c) maximize the monomer conversion (or minimize the reaction time),

 (d) minimize the polydispersity index, Q, of the polymer.

These objective functions and constraints are non-commensurate and are often conflicting in nature. The semi-batch free-radical copolymerization process, thus, is an excellent example of a multi-objective function optimization problem in the presence of constraints. For such problems, unique optimal solutions may not exist and we may obtain sets of several equally good (non-inferior) optimal solutions that constitute a

2

Pareto set [two non-inferior solutions are defined such that one solution has *any* one

objective function better (superior) than that of the other, but this is accompanied with a

worsening of *at least* one of the other objectives].[3,4] These performance objectives

compete against each other, and trade-offs among these need to be studied. The Pareto set

provides useful insights to a decision-maker, who can then use his judgment (higher-level

qualitative considerations)[4] to decide upon the preferred solution (operating point).

Indeed, Paretos are becoming an increasingly effective tool to determine the necessary

trade-offs between conflicting objectives.[4]

A simple technique used quite often for obtaining Pareto solutions is the weighted

sum approach, where a single objective function (a weighted sum of the several

objectives) is used. Several studies using this approach have been reported[5-7] in the 1980s

for homo-polymerizations. However, this technique has the disadvantage of hiding the

physical significance of the individual objectives, i.e., this technique gives the solution to

the overall problem but does not provide insights into the conflicts among the competing

objectives. Furthermore, it requires an *a priori* knowledge of the weights to be used for

each objective function, which is difficult to ascertain. These disadvantages are even

more critical for copolymerizations. In this case, several additional objectives are

important. For example, not all molecules formed have the same proportion of the

reacting monomers. In addition, there is the problem of the copolymer composition

distribution (CCD). Even for two copolymer molecules having the same overall ratio of

the monomer units, the possibility exists for different ordering of the comonomers along

the chain. This is the classic sequence distribution (SD) problem associated with

copolymerizations. Each of these distributions has its own unique and profound

3

influences on the properties of the product. Clearly, the control of SD is an equally important objective. The trade-offs among the several objectives are important, e.g., how much of an MWD objective has to be sacrificed in order to gain a certain improvement in the CCD? We attempt to answer a few such questions in this study.

The concept of multi-objective optimization is attributed to Pareto.[8] After several decades, this concept was recognized in operations research and has recently become popular in engineering. The first application of multi-objective optimization in the field of polymerization was presented by Tsoukas et al.[9] These workers studied the dual objectives: (a) narrowing the molecular weight distribution (minimizing Q of the final product), and (b) minimizing the deviation of the copolymer composition from a desired value at all times during the polymerization (minimize the integral of the square of the deviation). They used the ε-constraint algorithm[10] and Pontryagin's minimum principle.[11] Butala et al.[12] formulated a slightly different two-objective optimization problem for the copolymerization of styrene and acrylonitrile. They minimized two integrals: (a) the square of the deviation of the number average molecular weight of the copolymer at time, t, from a desired value, and (b) the square of the deviation of the composition of the unreacted monomers in the reaction mass (related to the copolymer composition) at time, t, from a desired value. They used a self-developed interactive code, CONSOLE (using a search technique with parameterized control variables), to obtain solutions for this problem as well as for some simpler optimization problems involving only a single objective function. Farber[13] reported the use of a similar multiobjective optimization technique (ε-constraint, followed by the iterative solution of the resulting Kuhn-Tucker problem[11]) to determine the non-inferior solutions for the continuous copolymerization of

4

the styrene-acrylonitrile and methyl methacrylate-vinyl acetate systems in CSTRs (continuous flow stirred tank reactors) operating under steady state. They studied three different pairs of objective functions selected from among: (a) maximize the number average molecular weight, (b) maximize the conversion of monomer 1, and (c) minimize the deviation of the copolymer composition from a desired value. They used the residence time and the temperature as the two decision variables (values, and not functions). Interestingly, multi-objective optimization was first brought to the field of chemical engineering through these studies in copolymerization. Lee et al.[14] used genetic algorithm (GA) to optimize the unsteady operation of CSTRs during grade change. They minimized the transition time required to produce copolymer having the desired values of the weight-average molecular weight and the fraction of vinyl acetate (VA) in the VA-methyl methacrylate (MMA) copolymer, again using an integral of the square of the deviations.

Most of the multi-objective optimization problems solved till now for copolymerization reactors have been relatively simple involving, at most, two of the important issues out of the four discussed earlier. Hence, there is still considerable scope for exploring the optimization of these reactors using more recent techniques, so as to obtain solutions for more complex and complete problems involving all the important objectives. In this work, we re-look at the multi-objective optimization of the solution copolymerization of styrene and acrylonitrile using an adaptation of genetic algorithm (GA). GA is an artificial intelligence (AI)-based technique and mimics the process of natural selection. It exploits the Darwinian principle of 'survival of the fittest' and offers several advantages over the earlier conventional techniques. GA, as first developed by

5

Holland[15] for single objective problems, has been adapted by several workers to solve problems involving multiple objectives. One such adaptation for problems involving decision variables that are single (optimal) *values*, is the non-dominated sorting genetic algorithm (NSGA-I) developed by Srinivas and Deb.[4,16] Mitra et al.[17] extended this technique to apply to problems where the decision variables are *functions* of time (trajectory-optimization problems). Unfortunately, NSGA-I has been observed to fail in some cases, usually described by complex equations, e.g., for PET reactors.[18] A modification of this algorithm, the *elitist* non-dominated sorting genetic algorithm (referred to as NSGA-II)[19] is now available in the literature that incorporates the concept of elitism in order to make the algorithm more powerful. This algorithm is being used for multi-objective optimization in the present study *after being adapted* to account for decision variables that are functions of time (trajectory optimization). Details of this adapted NSGA-II are provided in Appendix 1. Neither NSGA-I nor NSGA-II require good initial guess solutions (as for Pontryagin's minimum principle, etc.) and NSGA-II leads to globally optimal solutions (instead of getting stuck, at times, at local optima, as does NSGA-I). In the last few years, several studies have been reported on the multi-objective optimization of industrial units using either NSGA-I or NSGA-II. Examples in chemical engineering include the optimization of homo-polymerization reactors,[17,20,21] industrial steam reformers,[22,23] cyclone separators,[24] venturi scrubbers,[25] membrane separation modules,[26] fluidized-bed catalytic cracking (FCC) units,[27,28] etc. These have been reviewed recently by Bhaskar et al.[29] and Nandasana et al.[30] Use of elitism in NSGA-II leads to a rapid convergence to solutions near the optimal zone,[4] but, unfortunately, also leads to a reduction in the diversity of the solutions (gene pool), thus

introducing the possibility of premature convergence to false or local minima. This needs

to be compensated for. Macro-mutations can possibly achieve this.[28]

Modeling:

Any optimization study must be preceded with the development of a suitable model. Three types of important kinetic models have been reported in the open literature for copolymerizations. These are: (a) the penultimate effect model, (b) the diffusion model (accounting for the decrease in the rate constants with time, the Trommsdorff effect[32]), and (c) the ϕ-factor model.[32] The salient features of these models have been discussed by Odian.[32] The penultimate effect model accounts for the influence of the monomer unit next to the one incorporating the free radical at the end of the macro-radical, on the rate constants. It has considerable merit, but the experimental data required to estimate the rate parameters are usually not available in the open literature, except for a few systems. In addition, this model is quite complex and cumbersome to use. Similarly, expressions for the decreasing rate constants (propagation and termination) in the diffusion model are available for only a few systems. In contrast, the parameters for the ϕ-factor model are reasonably well documented, and so this model is quite popular for solution copolymerizations, where the Trommsdorff effect is weak.

In this work, the kinetic (ϕ-factor) model of Butala et al.[12] for the solution (in Xylene) polymerization of styrene and acrylonitrile (SAN) using AIBN as initiator, is used. This model has been validated using experimental data.[33] However, the general procedure for optimization developed herein is in no way restricted to this model of copolymerization alone, and we can use it with any more sophisticated model, or for any other copolymerization system.

The kinetic scheme for free radical copolymerization of SAN is given in Appendix 2, while the complete set of model equations[12] is shown in Appendix 3. The set of ordinary differential equations of the initial value kind (ODE-IVPs) in Appendix 3 can be written in the following general form:

$$dx/dt = F(x, u) \tag{1a}$$

$$x(t = 0) = x_o \tag{1b}$$

where x is the set of state variables describing the system, and u is the vector of control or decision variables. For the problem being studied, these are given by

$$x(t) = [M_1(t), M_2(t), I(t), V(t), \lambda_0(t), \lambda_1(t), \lambda_2(t)]^T \tag{2}$$

and

$$u(t) = [T(t), u(t)]^T \tag{3}$$

In Eqs. 2 and 3, M_1 and M_2 are the molar concentrations of the unreacted monomers, 1 (styrene) and 2 (acrylonitrile), respectively, I is the concentration of the unreacted initiator, V is the volume of the reaction mixture, λ_k ($k = 0, 1, 2, \ldots$) is the kth moment of the molecular weight distribution of the dead copolymer, T is the temperature of the well-mixed reaction mixture, and u is the volumetric rate of addition of the continuous feed of a monomer/initiator/solvent mixture (having a fixed composition) to the reactor. All of these are functions of the time, t, for the semi-batch reactor. No energy balance equation is required since T(t) is being used as a decision variable [note that T(t = 0) = T_o, the initial temperature of the material in the reactor].

The conversion of monomer 1 (styrene) at any time, t, for the semi-batch reactor can be defined as

$$x_1 = \frac{V_0 M_{10} + \int_0^t u(t) M_{1f} dt - VM_1(t)}{V_0 M_{10} + \int_0^t u(t) M_{1f} dt} \tag{4}$$

In this equation, subscripts 0 and f refer to the initial mixture taken in the semi-batch reactor, and the monomer mixture added continuously [at a rate, u(t)] to the reactor, respectively. An expression for the conversion of monomer 2 can be similarly written. The average mole fraction, F_1, of monomer 1 in the *dead* copolymer at time, t, is defined by the Mayo-Lewis equation[31]

$$F_1 = \frac{r_1\phi^2 + \phi}{r_1\phi^2 + 2\phi + r_2} \tag{5}$$

where ϕ indicates the composition of the mixture of *unreacted monomers* in the reaction mass:

$$\phi = \frac{M_1}{M_2} \tag{6}$$

and r_1 and r_2 are the reactivity ratios (defined in the Nomenclature). The values of the several rate constants for the SAN system are given in Table 1,[9] while the parameters describing the reactor system are given in Table 2.[12]

The model equations can be integrated for any specified set of *input* variables [namely, the amount and composition of the initial reaction mixture, the composition of the continuous feed stream, as well as T(t) and u(t)] using the D02EJF program in the NAG library. This code uses the Gear technique[34] for integrating stiff ODE-IVPs, and provides values of the *output* and state variables (e.g., the conversion, x_1, of monomer 1, the composition, ϕ, of the mixture of unreacted monomers, the number average molecular

Table 1: Kinetic Parameters for SAN Polymerization[9]

Solvent: Xylene; Initiator: AIBN; Monomer 1: Styrene; Monomer 2: Acrylonitrile

Parameter	Pre-exponential factor	Activation Energy, cal mol^{-1}
k_d	6.02×10^{15} s^{-1}	31730
f	0.6	--
k_{p11}	1.06×10^7 l mol^{-1}s^{-1}	7067
k_{tc11}	1.25×10^9 l mol^{-1}s^{-1}	1677
k_{f11}	2.31×10^6 l mol^{-1}s^{-1}	12670
k_{p22}	3.0×10^7 l mol^{-1}s^{-1}	4100
k_{tc22}	3.3×10^{12} l mol^{-1}s^{-1}	5400
k_{f22}	6.93×10^6 l mol^{-1}s^{-1}	5837
r_1	2.56	1190
r_2	6.67×10^{-5}	-4340
k_{f12}	$30 \times 2.31 \times 10^6$ l mol^{-1}s^{-1}	12670
k_{f21}	$5 \times 6.93 \times 10^6$ l mol^{-1}s^{-1}	5837
ϕ_t (for k_{tc12})	23	--

* Disproportionation rate constants = 0

11

Table 2: Parameters Characterizing the Semi-batch SAN Reactor[12]

	Parameter	Value
Initial Feed in Reactor	V_0	1.0 l
	$(M_1/M_2)_0$	1.0
	$M_{1,0}$	3.81 mol l^{-1}
	$M_{2,0}$	3.81 mol l^{-1}
	S_0	2.56 mol l^{-1}
	I_0	0.05 mol l^{-1}
	Solvent mole fraction	0.25
Continuous Feed Stream	$(M_1/M_2)_f$	1.5
	I_f	0.05 mol l^{-1}
	$M_{1,f}$	4.416 mol l^{-1}
	$M_{2,f}$	2.944 mol l^{-1}
	Solvent mole fraction	0.25
Reaction time	t_f	300 min

weight, M_n, of the copolymer, the polydispersity index, Q, and the state variables listed in

Eq. 2, etc., at any time, t). Some of the latter could be used directly as objective functions,

or could be used to evaluate them. This model is, then, used for solving a few multi-

objective optimization problems.

Optimization:

Several combinations of objective functions can be considered in any

optimization study of copolymerization processes. Since both the final average molecular

weight (and Q) and the average copolymer composition are related to the physical and

mechanical properties of the final product, these could possibly be used in some form,

either as objective functions or as end-point constraints. Also, we may wish to maximize

the monomer conversion so as to maximize the production. This could be taken as yet

another objective function (alternately, the classical time-minimization problem,

minimization of the total reaction time, t_f, could be solved). As mentioned earlier, the

sequence distribution (SD) of monomer units along the chain is important, but since the

present model does not provide any information on this quantity, we do not consider it as

an objective function in the present study. In this work, we start with the solution of a few

simple (two) multi-objective optimization problems for the SAN semi-batch reactor,

using different combinations of these objective functions/constraints. Once some intuition

is developed, we can use GA to obtain solutions of more complex problems than obtained

in earlier studies.[9,12,13] The latter would be of more relevance to real copolymerizations.

The first two-objective optimization problem (Problem 1) solved here is the same

as studied by Butala et al.[12] (Eqs. 43 and 44 in their paper; dimensionless form of the

objective function corresponding to M_n being used here). We minimize the time-averaged deviation of the mole ratio, ϕ, of the unreacted monomers from a desired value, ϕ_s (taken as 1.0 here). This ensures that the copolymer produced *at every t* during the reaction has (almost) similar composition, and in some manner, ensures a reasonably uniform sequence distribution (without really ensuring for this *exactly*). The time-averaged deviation of the (dimensionless) number-average molecular weight, M_n, is also minimized with respect to a desired value, $M_{n,s}$ (taken as[12] 30,000). This attempts to minimize, somewhat, the value of Q (while not ensuring this *exactly*), while simultaneously ensuring the production of product having a number average molecular weight close to the desired value. This problem can, thus, be written mathematically, as

Problem 1:

$$\text{Min } I_1(\mathbf{u}) = \int_0^{t_f} [\phi(t) - \phi_s]^2 \, dt \tag{6a}$$

$$\text{Min } I_2(\mathbf{u}) = \int_0^{t_f} \{[M_n(t) - M_{n,s}]/M_{n,s}\}^2 \, dt \tag{6b}$$

subject to (s.t.):

$$323 \leq T(t) \leq 368 \text{ (K)} \tag{6c}$$

$$0.0 \leq u(t) \leq 0.07 \text{ (1 min}^{-1}) \tag{6d}$$

Clearly, no emphasis is being given to the attainment of high conversions in this first problem, primarily to keep it simple. Also, no *explicit* attempt is made to obtain a polymer having a narrow distribution of molecular weight (low value of Q). The total reaction time, t_f, is assumed to be specified (= 300 min).[12] The range of values of the decision variables, namely, the temperature, T(t), and the rate of addition, u(t), of the

continuous feed are given above, and are the same as used by Butala et al.[12] Special (one-objective function) cases of this problem (also solved by Butala et al.), provide excellent means of testing out our multi-objective optimization code (using all the objectives in the code to be the same).

The second optimization problem studied here involves two objective functions that are the same as those used by Tsoukas et al.[9] The complete problem is described by:

Problem 2:

$$\text{Min } I_1(\mathbf{u}) = \int_0^{t_f} [F_1(t) - F_{1,s}]^2 \, dt \tag{7a}$$

$$\text{Min } I_2(\mathbf{u}) = Q(t_f) \tag{7b}$$

s.t.:

$$323 \le T(t) \le 368 \ (K) \tag{7c}$$

$$0.0 \le u(t) \le 0.07 \ (l \ min^{-1}) \tag{7d}$$

Here, $F_1(t)$ is the mole fraction of the monomeric units, 1, in the *dead* copolymer chains at time t, while $F_{1,s}$ (= 0.619) is its desired value, taken to be the same as in Tsoukas et al.[9] The bounds on the two decision variables, the value of t_f (= 300 min), the amount and composition of the initial feed, and the composition of the continuous feed stream, are those assumed in Problem 1 (the same as those used by Butala et al.[12]). This has been done so that the results obtained for Problems 1 and 2 can be compared. It may be mentioned that this problem does not require the production of copolymer having a desired value of M_n, nor does it attempt to maximize the monomer conversion. Thus, Problems 1 and 2 are not the most complete multi-objective optimization problems for copolymer reactors, since they address only some of the important parameters. More

meaningful problems should address all four important issues relevant in copolymer manufacture: maximization of the monomer conversion (for a specified value of t_f, or minimization of t_f), obtaining polymer product having a desired value of M_n with the lowest value of Q, and producing copolymer of a desired composition. Two such multi-objective optimization problems are now described. These are much more constrained and so are difficult to solve for (the windows of the solutions are narrower). These have not been solved earlier, either.

The first of these problems using only two objective functions (to keep it simple), is given by:

Problem 3 (Reference Problem):

$$\text{Max } I_1(\mathbf{u}) = \quad x_1(t_f) \tag{8a}$$

$$\text{Min } I_2(\mathbf{u}) = \quad Q(t_f) \tag{8b}$$

s.t.:

$$M_n(t_f) \equiv M_{n,s} \pm TOL_M = 30{,}000 \pm 500 \tag{8c}$$

$$F_1(t_f) \equiv F_{1,s} \pm TOL_F \quad = 0.600 \pm 0.005 \tag{8d}$$

$$323 \leq T(t) \leq 368 \ (K) \tag{8e}$$

$$0.0 \leq u(t) \leq 0.07 \ (1 \ \text{min}^{-1}) \tag{8f}$$

The value of $F_{1,s}$ used in this problem is different from the value of 0.619 used[9] in the less-constrained Problem 2, and it was not possible to obtain solutions using the latter value of $F_{1,s}$ along with the molecular weight constraint (the initial conditions, etc., used by Tsoukas et al.[9] were different than the values being used in Problems 1-3). The bounds on the two decision variables, the value of t_f (= 300 min), the amount and composition of the initial feed, and the composition of the continuous feed stream, are the same as in

Problems 1 and 2.[12] The tolerances on $M_{n,s}$ and $F_{1,s}$ specify narrow ranges, beyond which penalties are imposed, as described in Appendix 1.

Still better optimization problems for copolymer reactors would involve three (or more, though this would make interpretation of the results almost impossible) objective functions. The results of such problems are difficult to visualize, but one such problem (Problem 4) is being presented here, to illustrate the versatility of GA:

Problem 4:

$$\text{Max } I_1(\mathbf{u}) = x_1(t_f) \tag{9a}$$

$$\text{Min } I_2(\mathbf{u}) = Q(t_f) \tag{9b}$$

$$\text{Min } I_3(\mathbf{u}) = \int_0^{t_f} [\phi(t) - \phi_s]^2 \, dt \tag{9c}$$

s.t.:

$$M_n(t_f) = M_{n,s} \pm \text{TOL}_M = 30{,}000 \pm 500 \tag{9d}$$

$$323 \le T(t) \le 368 \ (\text{K}) \tag{9e}$$

$$0.0 \le u(t) \le 0.07 \ (l\,\text{min}^{-1}) \tag{9f}$$

This problem addresses all four of the important issues in the optimization of copolymer reactors, and incorporates the best of the previous three problems. The bounds on the two decision variables, the value of t_f (= 300 min), the amount and composition of the initial feed, and the composition of the continuous feed stream, are, once again, the same as used[12] in Problem 1. Table 3 summarizes some details of these four problems. It may be mentioned that minimization of t_f [with an end point constraint of achieving a desired value of $x_1(t_f)$] is not attempted here since it is similar to the maximization of $x_1(t_f)$ with t_f specified.

Table 3: Summary of Optimization Problems Studied

	Problem 1	Problem 2	Problem 3	Problem 4
ϕ_s	1.0	--	--	1.0
$M_{n,s}$, kg kmol^{-1}	30000	--	30000	30000
t_f , min	300	300	300	300
$F_{1,s}$	--	0.619	0.600	--
TOL_M	--	--	500	500
TOL_F	--	--	0.005	--

RESULTS AND DISCUSSION

In this study, we use the *elitist* non-dominated sorting Genetic Algorithm adapted for continuous decision variables, to solve the several multi-objective optimization problems described above. The code used actually maximizes *all* the objective functions. This poses no problem for objective functions that are to be minimized. However, such objective functions that are to be minimized, are converted to fitness functions, F [using $F = 1/(1 + I)]^{3,4}$ that are then maximized. For maximization of any objective, the fitness function and objective functions are identical. The end point constraints in some of the problems are taken care of by using a penalty[3,4,11] approach. If any chromosome (solution) generated by NSGA-II does not satisfy the constraint within the tolerance specified, the value of the fitness function is artificially made very small (10^{-4}), so that this chromosome is killed quickly.

A computer code was written in Fortran 90 to obtain the results of the multi-objective optimization problems described earlier, using NSGA-II. The CPU time required for solving a typical problem, e.g., the Reference problem (Problem No. 3), using a population size of 50 chromosomes and computing till the 150th generation, is 2.7 hr on a Silicon Graphics Server (SGI Origin 200, 270 MHz, IRIX 6.5 Operating System). The code was first tested in the simulation mode using a single chromosome, with specified T(t) and u(t) [two examples described by Butala et al.[12]: isothermal with u(t) = 0; non-isothermal with specified u(t)], and the results were found to match those reported earlier.[12] The NSGA-II code was then tested against a few standard optimization problems,[3,4] both for single as well as for two and three objective functions. Several other

checks were also made to ensure that the code was free of errors before using it for generating results for the SAN reactor. The computational parameters used in NSGA-II for all these problems are given in Table 4.

Fig. 1 shows the results of Problem 1. Both the objective functions [I_1, the cumulative deviation of the mole ratio of the unreacted monomers, and, I_2, the cumulative deviation of the dimensionless number average molecular weight from their desired values] are to be minimized in this problem. Fig. 1a shows that one of the objective functions can be improved (decreased) only when the other worsens (increases). Thus, we obtain a Pareto set of non-dominant optimal solutions. More detailed investigations (the evolution of the Pareto set over the generations; results not shown here for brevity) show that the results for the 100th generation do not differ much from those at the 150th generation, and one can reduce the computational time easily, at least for *this* simple problem. Figs. 1b-e show the values of some additional (and important) variables describing the state of the product stream, namely, the number average molecular weight, $M_n(t_f)$, of the product copolymer, the average mole fraction, $F_1(t_f)$, of monomeric unit, 1, in the copolymer produced, the conversion, $x_1(t_f)$, of monomer 1, at the end of the reaction, and the polydispersity index, $Q(t_f)$, of the product, for all the non-dominating optimal chromosomes at the 150th generation. Of these four, one, namely, $x_1(t_f)$, is not focussed upon *at all* in Problem 1 (the other three are at least influenced indirectly or directly because of the choice of the two objectives). The optimal decision variable, u(t), for three chromosomes, A, B and C (shown in Fig. 1a), are shown in Fig. 2. It may be noted here that Butala et al.[12] do not provide the *complete* Pareto set, and provide detailed histories only for one of the (unspecified) Pareto-optimal solutions. This is probably

Table 4: Values of the NSGA-II parameters[3,4] used in this study

Parameter	Value
No. of binaries per substring (l_{str})	10
Maximum number of generations (N_{gen})	150
Population size (N_p)	50
Number of discrete points generated for each decision variable in the range 0 to t_f (N_{ga})	10
Crossover probability (p_c)	0.98
Mutation probability (p_m)	0.0
Random seed number (N_{seed})	0.8765
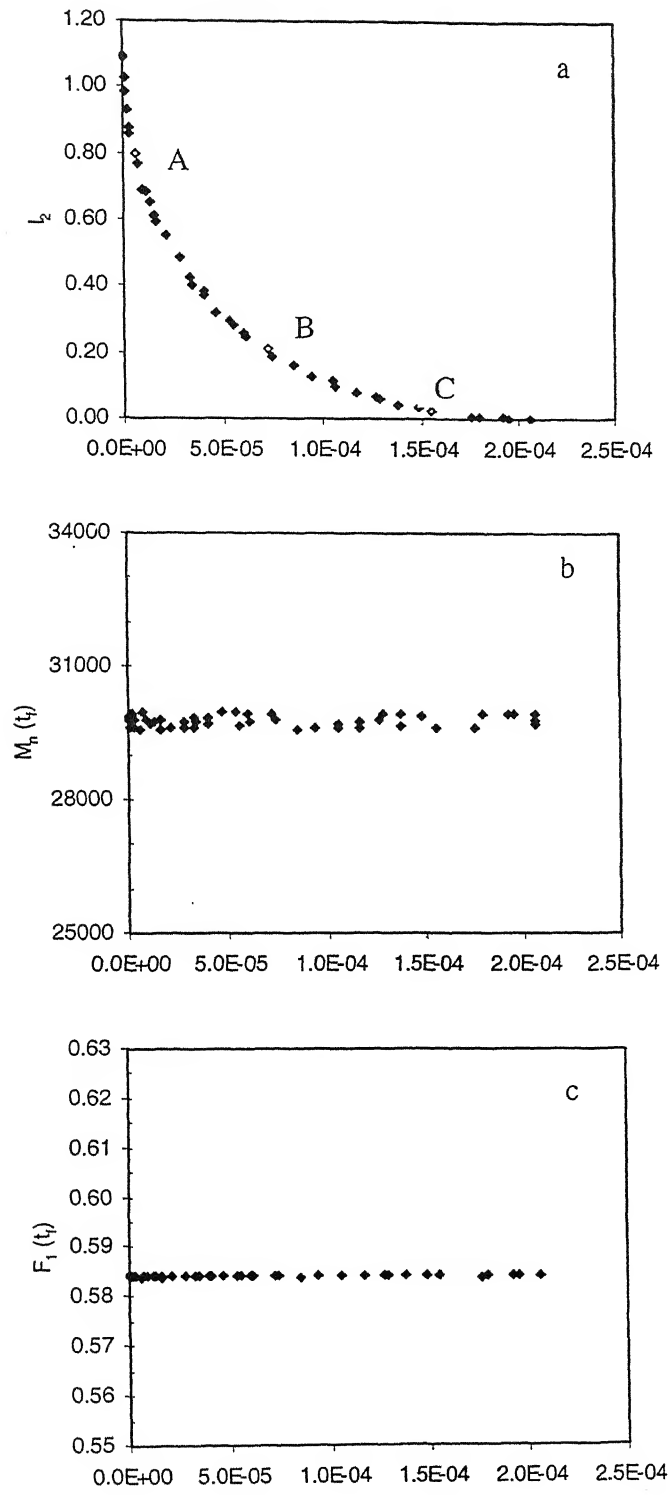
Figure 1: Pareto set and state variables of optimal solutions for Problem 1

22
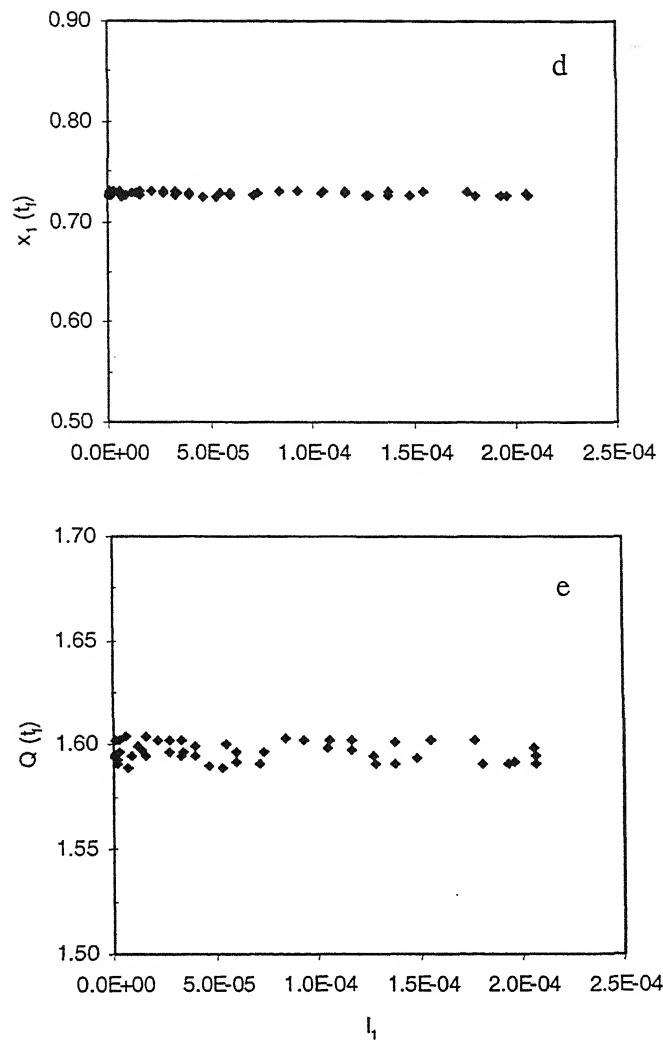
Figure 1 (contd.): Pareto set and state variables of optimal solutions for Problem 1
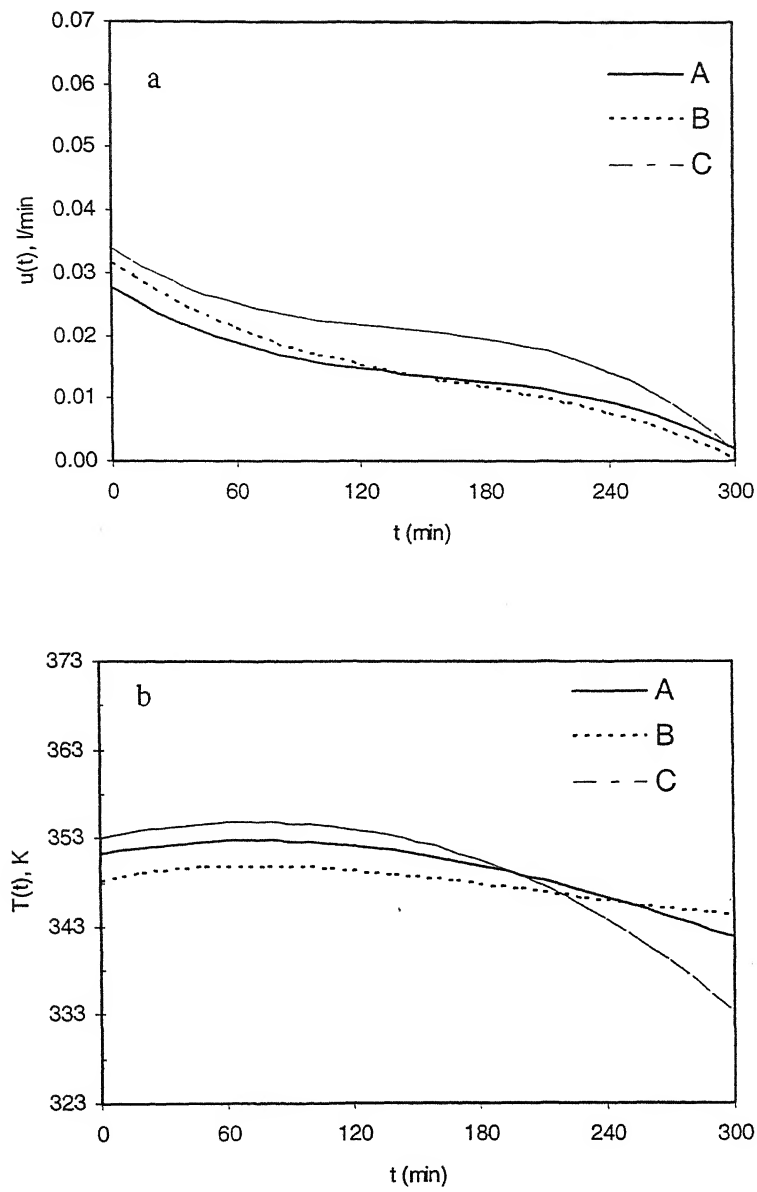
Figure 2: Optimal control variable histories for the three chromosomes A, B and C (shown in Fig. 1a) for Problem 1

because the optimal values of both the objective functions (Fig. 1a) are very low (i.e., all the Pareto points are indistinguishable, for all practical purposes). Indeed, the three histories shown in Fig. 2 are almost indistinguishable.

Figs. 3 and 4 show the results for Problem 2. Once again, Pareto solutions are obtained (Fig. 3a), as was obtained by Tsoukas et al.,[9] who solved the same problem but under different (input) conditions. Lower conversions (approximately 50%) are obtained for this case, compared to Problem 1. The values of I_2 are quite small, and this shows up in the small amount of scatter in Fig. 3c. However, there is considerable scatter in both $M_n(t_f)$ and $x_1(t_f)$.

The most interesting, two-objective optimization problem that addresses almost all the important issues for copolymer reactors, is Problem No. 3. This is why we refer to this problem as the *reference* problem, and study it in great detail. This problem is far more constrained than the previous ones, because of which the window of solutions is relatively narrower, and obtaining the 'correct' optimal solutions requires a considerable amount of exploration of the several computational parameters of NSGA-II. Indeed, the parameters given in Table 4 have actually been obtained for this problem, and then used for solving the other problems. Figs. 5 and 6 show the results. A Pareto set of solutions is obtained again (Fig. 5a). It is also observed (Figs. 5b and c) that *all* the chromosomes (except a few) satisfy the narrow range of values specified for $M_n(t_f)$ and $F_1(t_f)$ [$M_n(t_f)$ = 30,000 ± 500; $F_1(t_f)$ = 0.600 ± 0.005]. The few chromosomes violating these constraints are found to be generated in the previous (149th) generation. Increasing the number of generations would eliminate or improve these points, but at the cost of increased computational costs. One can easily disregard these outliers. The plots of the control
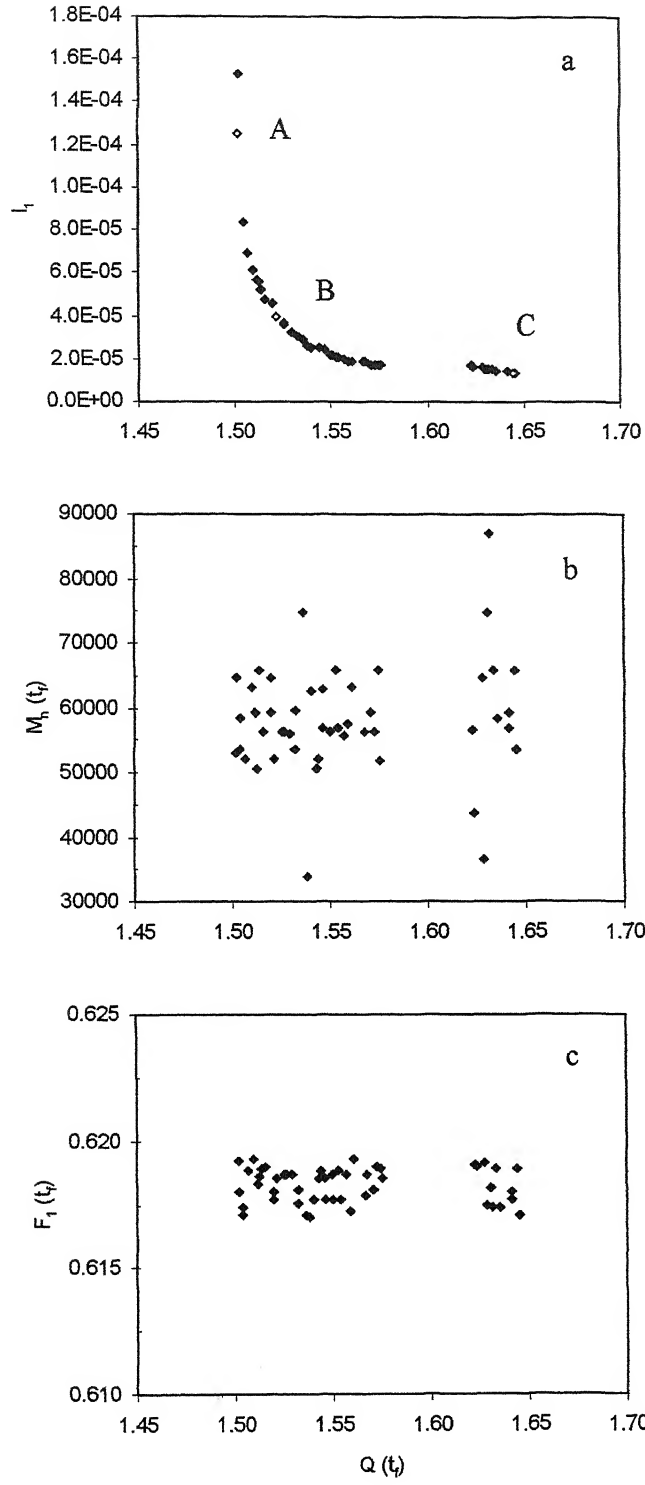
Figure 3: Pareto set and state variables of optimal solutions for Problem 2
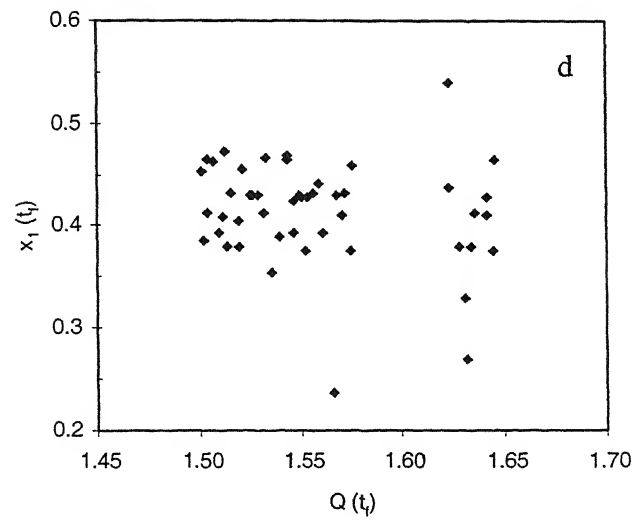
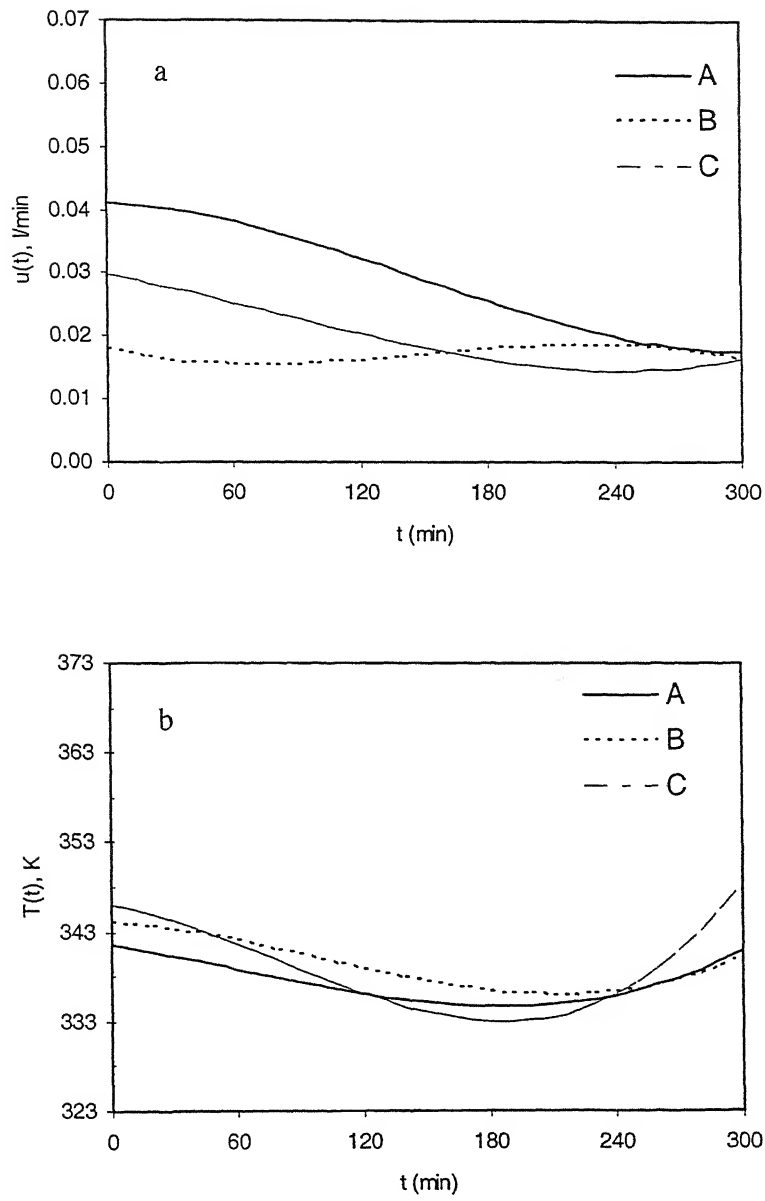Figure 3 (contd.): Pareto set and state variables of optimal solutions for Problem 2

Figure 4: Optimal control variable histories for the three chromosomes A, B and C (shown in Fig. 3a) for Problem 2
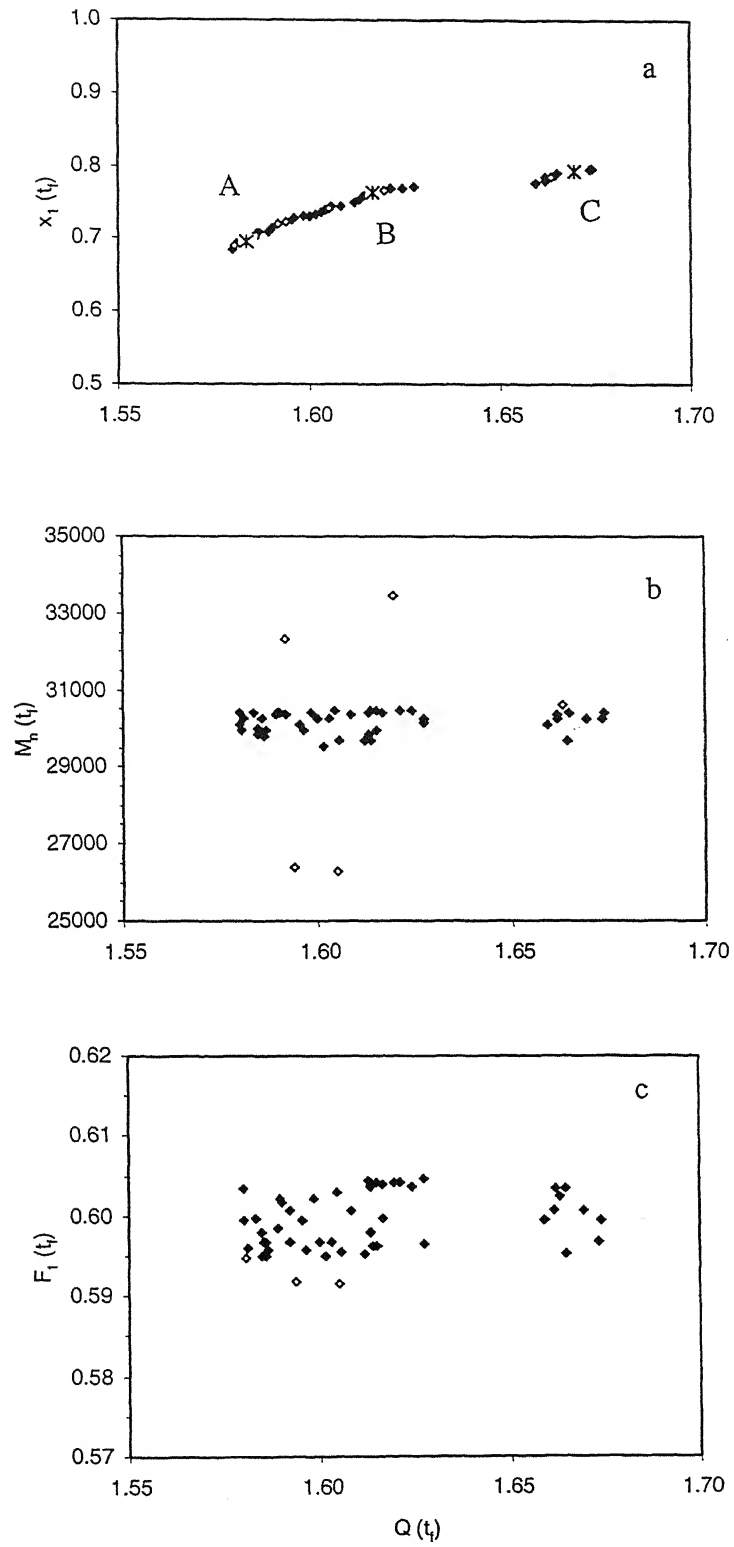
28

Figure 5: Pareto set and state variables of optimal solutions for Problem 3 (unfilled points denote chromosomes violating constraints)
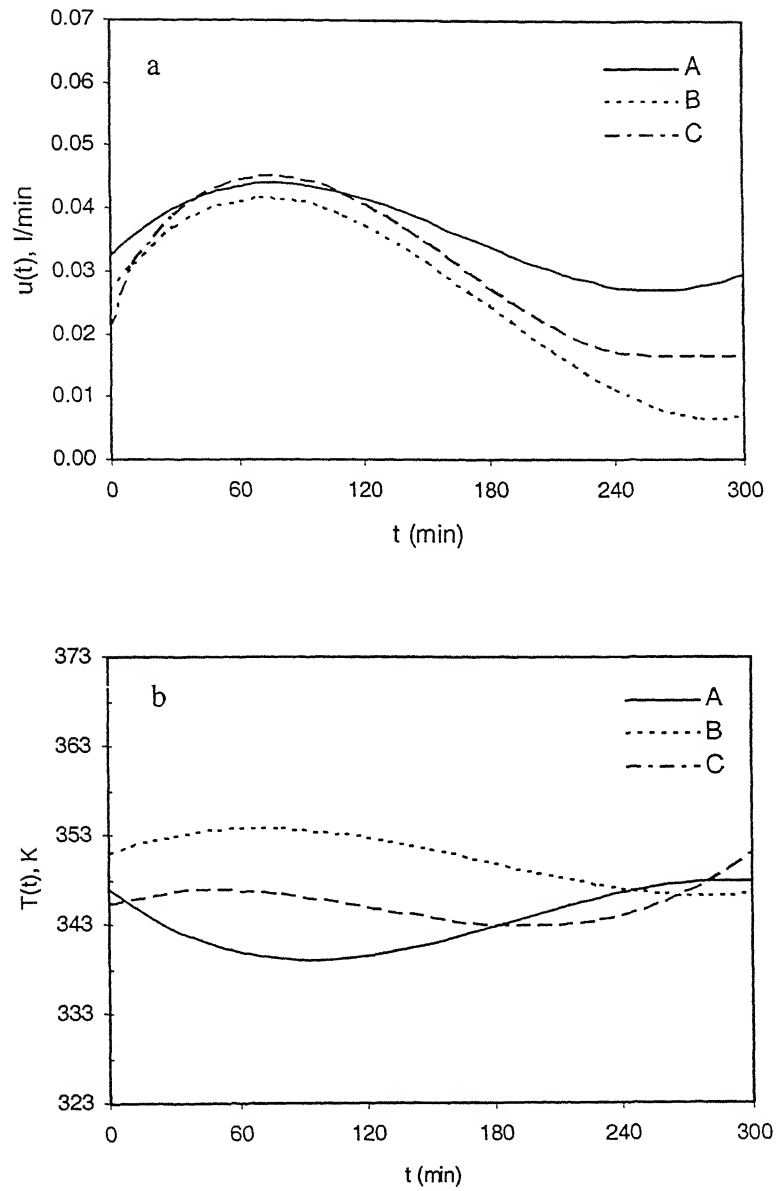
Figure 6: Optimal control variable histories for the three chromosomes A, B and C
(shown in Fig. 5a) for Problem 3

variables for three chromosomes, A, B and C (Fig. 5a) are given in Fig. 6. The higher

values of $x_1(t_f)$ compared to Problem 2, are to be noted.

Problem 4 involves three objective functions and an end-point constraint. In order

to interpret the optimal solutions, we first sort the 50 Pareto chromosomes in the order of

increasing values of $x_1(t_f)$. Fig. 7a shows this increasing monomer conversion as a

function of the chromosome number (re-numbered after such sorting). Fig. 7b shows the

second objective function, $Q(t_f)$, for these sorted chromosomes. Interestingly, an increase

in $x_1(t_f)$ (improvement) is associated with an increase (worsening) of $Q(t_f)$, which is not

surprising for an addition type polymerization. We can see the characteristics of a Pareto

right away, without even looking at the results of the third objective function, which is

important for copolymerization systems. The values of I_3 for these chromosomes are

observed to first decrease (improve) and then increase (worsen), as observed in Fig. 7c. It

may be emphasized at this point that Problem 3 (involving objective functions similar to

I_2 and I_3 in Problem 4) leads to a Pareto set, and that one need not obtain Paretos *only*

when using $x_1(t_f)$ and $Q(t_f)$ as objective functions. The interplay of the objective functions

is quite interesting. Fig. 7d shows excellent constraint satisfaction for Problem 4. Fig. 7e

shows the composition, $F_1(t_f)$, of the final copolymer (no constraints put on this, but it is

controlled indirectly through I_3). In contrast to Problem 1, the optimal values of I_3 for this

3-objective problem are higher, and the values of $F_1(t_f)$, exhibit greater scatter. Fig. 8

shows the optimal histories of the two decision variables for three chromosomes shown

in Fig. 7a. Introduction of an *additional* end-point constraint [$F_1(t_f) \equiv F_{1,s} \pm TOL_F =$

0.600 ± 0.005] in Problem 4 gives (results not provided here) *qualitatively* similar results

for $x_1(t_f)$ and $Q(t_f)$, but the value of I_3 falls continuously with the chromosome number
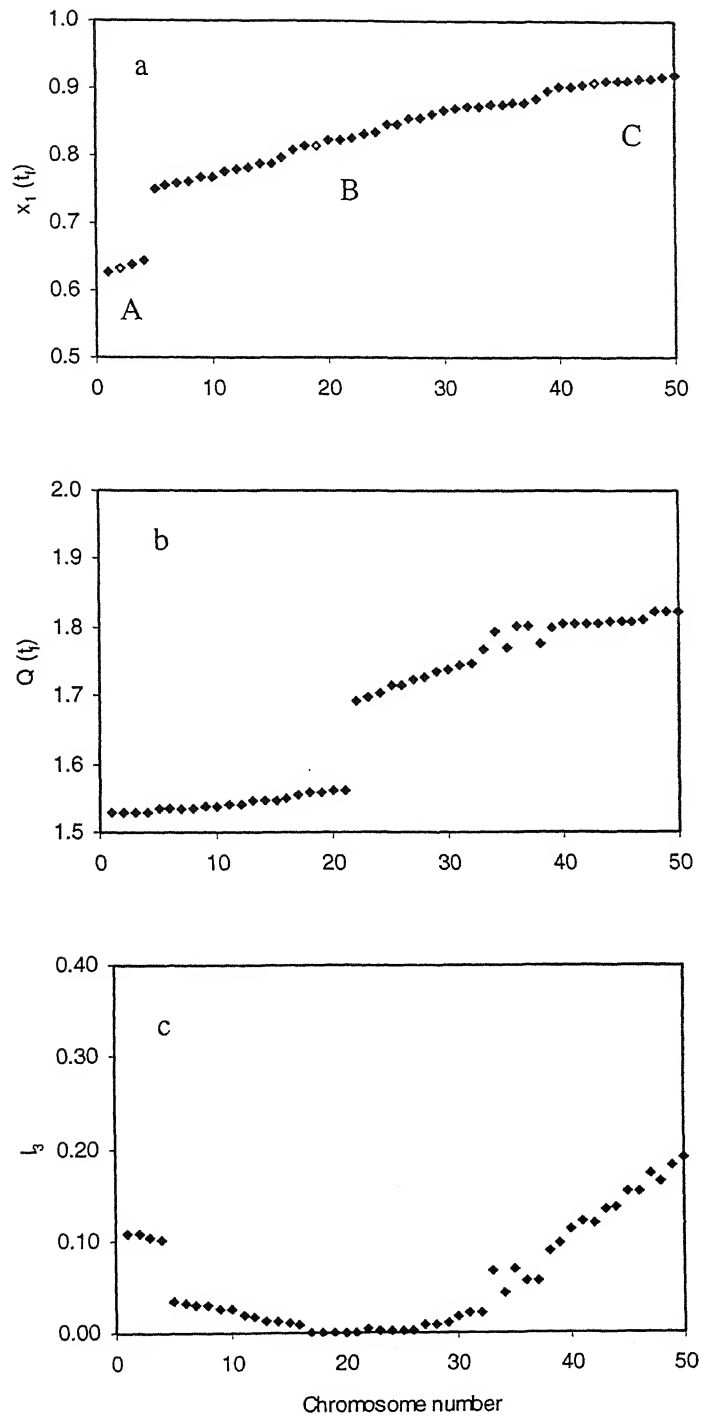
Figure 7: Plots of objective functions and state variables against chromosome number for Problem 4
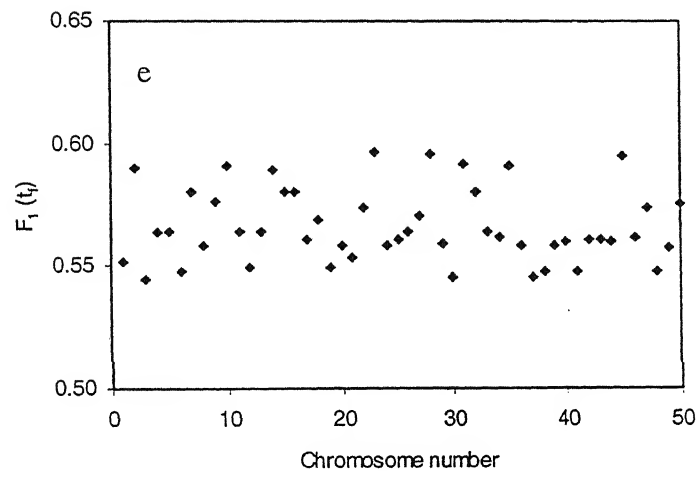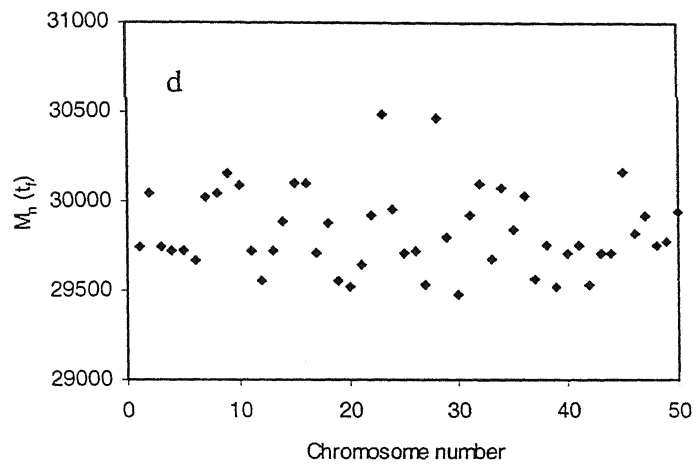
Figure 7 (contd.): Plots of objective functions and state variables against chromosome number for Problem 4
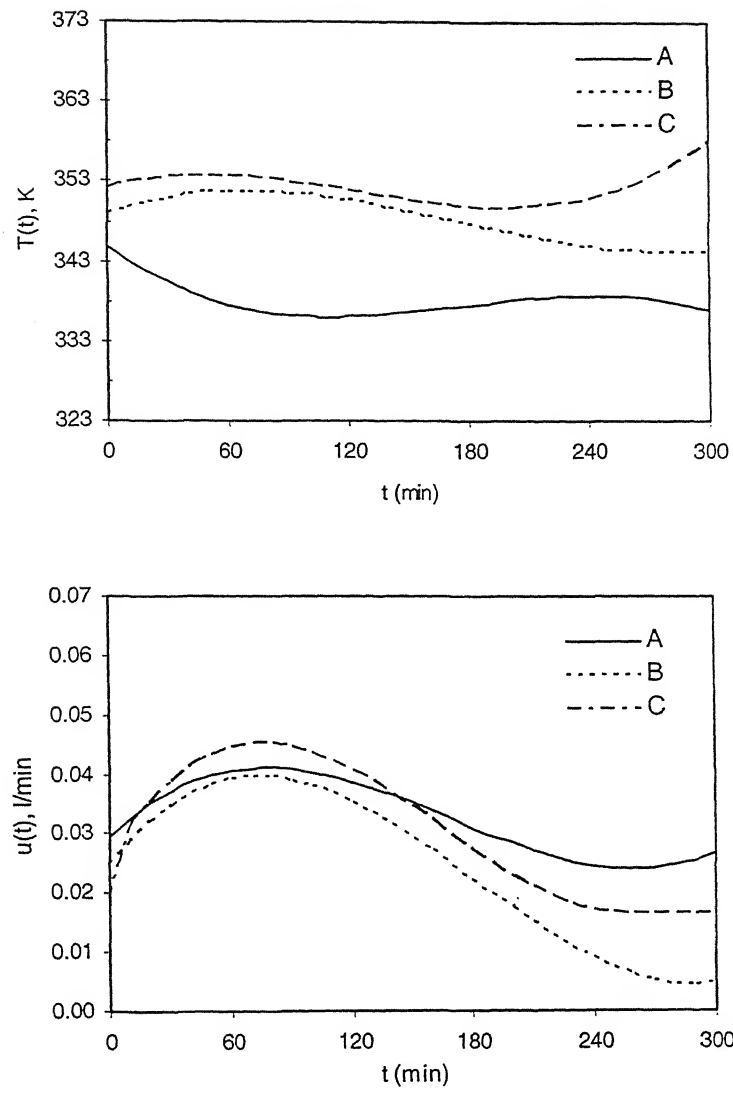
Figure 8: Optimal control variable histories for the three chromosomes A, B and C (shown in Fig. 7a) for Problem 4

(i.e., the final conversion and the integrated deviation of ϕ both improve as the polydispersity index of the product worsens). The scatter of $F_1(t_f)$ around the specified value becomes much less in this case.

The effect of varying the computational parameters, p_m, p_c, the random seed, and the number of generations, on the results of Problem 3, is now studied (not all the results are being shown here). It is found that an increase of p_m from 0 to 0.01 does not alter the results too much, but a further increase of p_m to 0.02 significantly worsens the constraint violations, particularly of $F_1(t_f)$. This is accompanied by a worsening of the Pareto as well. The spread of the Pareto is worsened by changing the value of p_c to 0.99 as well as to 0.90, while not affecting either of the constraint violations much. A similar thing is observed when the random seed is changed from 0.8765 to either 0.7654 or 0.8888. These results are not being provided here for the sake of brevity, and can be provided on request. Fig. 9 shows the development of the results (Pareto, as well as the constraint violations) over the generations. The Pareto set starts to appear in about the 10th generation, and there is almost no change after the 100th generation. In fact, results for the 50th generation are quite acceptable, even though there is a small amount of improvement in the Pareto as the number of generations increases beyond 50. Similarly, both the constraint violations in the 50th generation appear to be acceptable, though some improvements do occur after this.

Several additional multi-objective optimization problems can also be formulated and solved using NSGA-II as adapted herein for trajectory optimization cases. However, in this work we have presented solutions for a few problems that are more meaningful than those studied earlier.
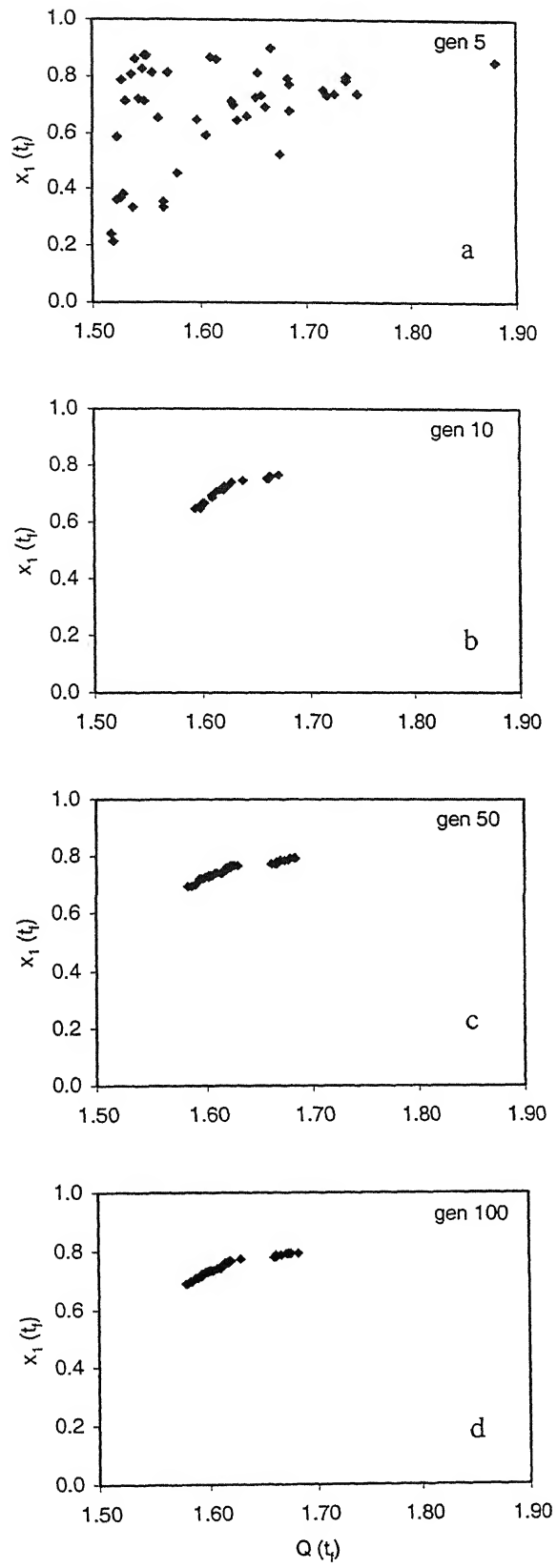
35

Figure 9: Evolution of Pareto and state variables over generations for Problem 3
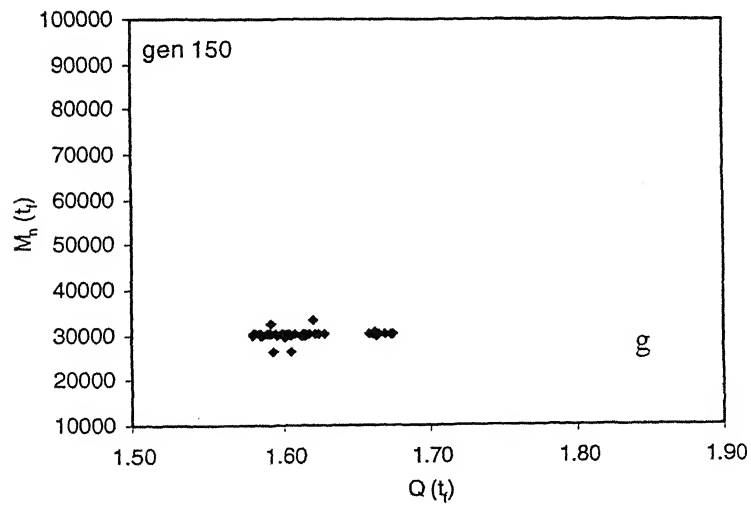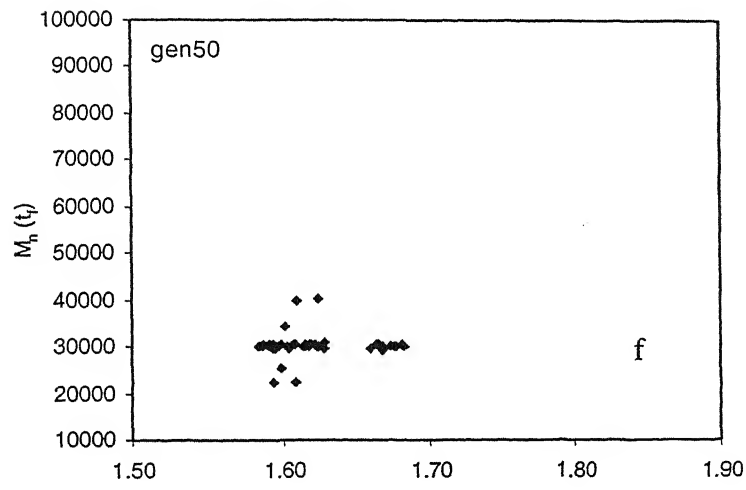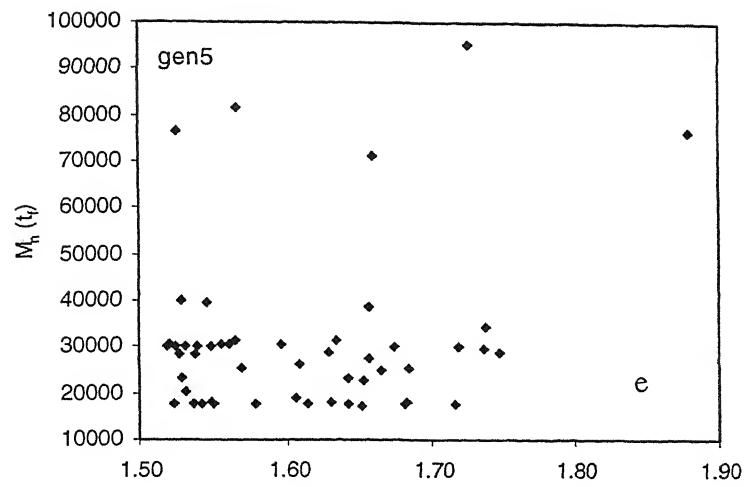
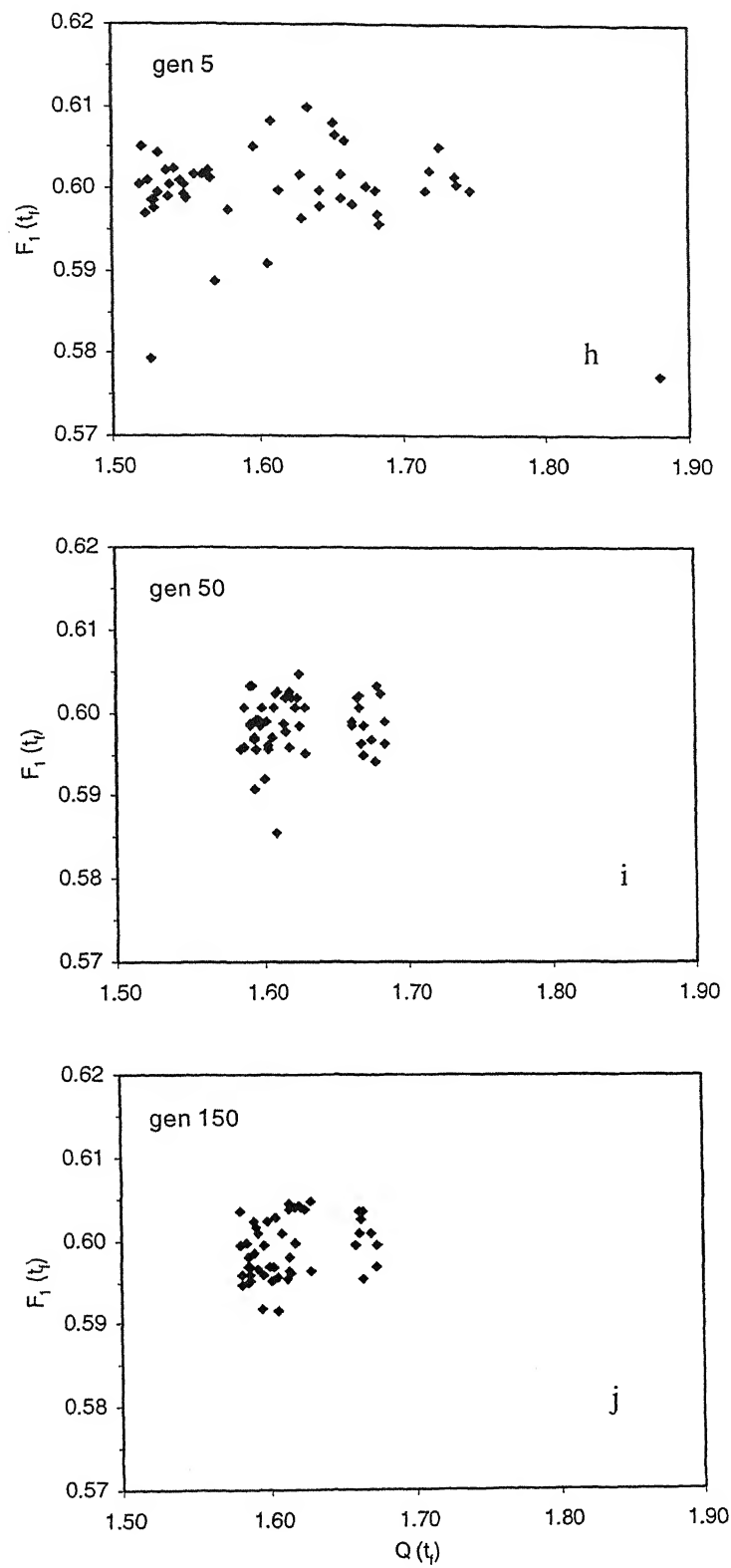Figure 9 (contd.): Evolution of Pareto and state variables over generations for Problem 3

Figure 9 (contd.): Evolution of Pareto and state variables over generations for Problem 3

38

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

Conclusions:

Multi-objective optimization of SAN polymerization in a semi-batch reactor is studied in this work. More complex problems involving four important properties relevant to copolymerization systems have been formulated and solved using adapted NSGA-II, than considered earlier in the literature. The technique is general and can be used for other copolymerizations as well.

Recommendations for Future Work:

1. The model used in this work is for solution copolymerization. A more rigorous model for bulk polymerization, considering the Trommsdorff effect or *'gel effect'*, could be used.

2. The optimization technique developed here can be extended for continuous reactors (CSTRs) also.

3. The sequence distribution of the dead copolymer can be optimized to produce a specified type of copolymer (e.g., block or alternating copolymers).

4. This work can be extended for the implementation of on-line inferential optimizing control and development of software sensors in copolymerization processes.

REFERENCES

1. R.B. Mankar, D.N. Saraf, and S.K. Gupta, *J. Appl. Polym. Sci.*, **85**, 2350 (2002).

2. G. E. Molau, *Polym. Lett.*, **3**, 1007 (1965).

3. K. Deb, *Optimization for Engineering Design: Algorithms and Examples*, Prentice Hall of India, New Delhi, India, 1995.

4. K. Deb, *Multiobjective Optimization using Evolutionary Algorithms*, Wiley, Chichester, UK, 2001.

5. J. N. Farber, and R. L. Laurence, *Chem. Eng. Commun.*, **46**, 347 (1986).

6. I. M. Thomas, and C. Kiparissides, *Can. J. Chem. Eng.*, **62**, 284 (1984).

7. S. R. Ponnuswamy, Online measurements and control of a batch polymerization reactor, Ph.D. Thesis, University of Alberta, Edmonton, 1984.

8. V. Pareto, *Cours d' Economie Politique*, Lausanne, Switzerland, 1896.

9. A. Tsoukas, M. Tirrell, and G. Stephanopoulos, *Chem. Eng. Sci.*, **37**, 1785 (1982).

10. V. Chankong, and Y. Y. Haimes, *Multiobjective Decision Making Theory and Methodology*, Elsevier, New York, 1983.

11. T. F. Edgar, and D. M. Himmelblau, *Optimization of Chemical Processes*, 2nd ed., McGraw Hill, New York, 2001.

12. D. Butala, K. Y. Choi, and M. K. H. Fan, *Comp. Chem. Eng.*, **12**, 1115 (1988).

13. J. N. Farber, *Polym. Eng. Sci.*, **26**, 499 (1986).

14. M. H. Lee, C. Han, and K. S. Chang, *Comp. Chem. Eng.*, **21**, S1037 (1997).

15. J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.

16. N. Srinivas, and K. Deb, *Evolutionary Comput.*, **2**, 106 (1995).

17. K. Mitra, K. Deb, and S. K.Gupta, *J. Appl. Polym. Sci.*, **69**, 69 (1998).

18. V. Bhaskar, S. K. Gupta, and A. K. Ray, *Comp. Chem. Eng.*, **25**, 391 (2001).

19. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, *IEEE Trans. Evolutionary Comp.*, **6**, 182 (2002).

20. V. Bhaskar, S. K. Gupta, and A. K. Ray, *AIChE J.*, **46**, 1046 (2000).

21. F. Zhou, S. K. Gupta, and A. K. Ray, *J. Appl. Polym. Sci.*, **78**, 1439 (2000).

22. J. K. Rajesh, S. K. Gupta, G. P. Rangaiah, and A. K. Ray, *Ind. Eng. Chem. Res.*, **39**, 706 (2000).

23. A. D. Nandasana, A. K. Ray, and S. K. Gupta, *Ind. Eng. Chem. Res.*, submitted.

24. G. Ravi, S. K. Gupta, and M. B. Ray, *Ind. Eng. Chem. Res.*, **39**, 4272 (2000).

25. G. Ravi, S. K. Gupta, S. Viswanathan, and M. B. Ray, *Ind. Eng. Chem. Res.*, **41**, 2988 (2002).

26. C. C. Yuen, Aatmeeyata, S. K. Gupta, and A. K. Ray, *J. Membrane Sci.*, **176**, 177 (2000).

27. R. B. Kasat, D. Kunzru, D. N. Saraf, and S. K. Gupta, *Ind. Eng. Chem. Res.*, **41**, 4765 (2002).

28. R. B. Kasat, and S. K. Gupta, *Comp. Chem. Eng.*, submitted.

29. V. Bhaskar, S. K. Gupta, and A. K. Ray, *Reviews Chem. Eng.*, **16**, 1 (2000).

30. A. D. Nandasana, A. K. Ray, and S. K. Gupta, *Int. J. Chemical Reactor Eng.*, submitted.

31. L. Stryer, *Biochemistry,* 4th ed., W. H. Freeman, New York, 2000.

32. G. Odian, *Principles of Polymerization*, 3rd ed., Wiley-Interscience, New York, 1991.

33. K. Y. Choi and D. N. Butala, *Polym. Eng. Sci.,* **31**, 353 (1991).

34. S. K. Gupta, *Numerical Methods for Engineers*, Wiley Eastern/New Age Intl., New Delhi, India, 1995.

Appendix 1

Elitist Non-dominated Sorting Genetic Algorithm
NSGA-II adapted for trajectory optimization[17] (see Flowchart in Fig. A1.1)

<u>NOTE:</u> The following assumes that we are maximizing *all* the fitness functions, F_q

1. Generate box, P, of N_p parent chromosomes using a random-number code to generate the several binaries. These chromosomes are given a sequence (position) number as generated. Each chromosome carries information on the *discretized* decision-variable *histories* [i.e., each decision variable history is discretized and represented in terms of N_{ga} equi-spaced points in $0 \le t \le t_f$]

2. Map the binary coded chromosomes.[3] The range in which the *first* binary for any decision variable is mapped is as specified externally (e.g., Eqs. 6c and d). The remaining binaries for that decision variable are mapped adaptively in the range, $\mathbf{u}_{i+1} = \mathbf{u}_i \pm \Delta \mathbf{u}_i$, where $\Delta \mathbf{u}_i = 5$ K for T(t), and $\Delta \mathbf{u}_i = 0.01$ l min^{-1} for u(t). The adaptively mapped, discretized values of each decision variable are curve-fitted to obtain continuous functions, using a Chebyshev polynomial fitting subroutine (E02ACF from the NAG FORTRAN library). The model equations are then solved and the values of the fitness functions are evaluated. If any of the end-point constraints is violated, the fitness function is artificially reduced to a very low value (10^{-4} in this work) so that they can get killed soon

3. Classify these chromosomes into *fronts* based on non-domination,[3,4] as follows:

   a) Create a new (empty) *box*, P', of size, N_p

   b) Transfer ith chromosome from P to P', starting with i = 1

   c) *Compare chromosome i with each* member, say, j, already present in P', one at a time

   d) If i **dominates**[4] over j (i.e., i is superior to or better than j in terms of *all* objective functions), remove the jth chromosome from P' and it put it back in its original location in P

   e) If i is dominated over by j, remove i from P' and put it back in its position in P

42

f) If i and j are **non-dominating** (i.e., there is *at least one* objective function associated with i that is superior to /better than that of j), keep both i and j in P' (in sequence). Test for all j present in P'

g) Repeat for the next chromosome (in sequence, without going back) in P till all N_p are tested. P' now contains a sub-box (of size $\leq N_p$) of non-dominated chromosomes (a subset of P), referred to as the first *front* or sub-box. Assign it a rank number, I_{rank}, of 1

h) Create subsequent fronts in (lower) sub-boxes of P', using Step 2b above (with the chromosomes remaining in P). Compare these members *only* with members present in the *current* sub-box, and not with those in earlier (better) sub-boxes. Assign these I_{rank} = 2, 3, ... Finally, we have all N_p chromosomes in P', boxed into one or more fronts

i) Assign very low fitness values to chromosomes violating end-point constraints

4. **Spreading out**: Evaluate the crowding distance, $I_{i,\,dist}$, for the ith chromosome in any front, j, of P' using the following procedure:

   a) Rearrange *all* chromosomes in front j in ascending order of the values of *any one* (say, the qth) of their several objective functions (fitness functions). This provides a sequence, and, thus, defines the nearest neighbors of any chromosome in front j

   b) Find the largest cuboid (rectangle for two fitness functions) enclosing chromosome i that just touches its nearest neighbors in the f-space

   c) $I_{i,dist} \equiv \frac{1}{2} \times$ (sum of *all* sides of this cuboid)

   d) Assign large values of $I_{i,dist}$ to solutions at the boundaries (the convergence characteristics would be influenced by this choice)

5. Make N_p copies randomly (duplication permissible), of the better chromosomes from P' into a new box, P'' using:

   a) Select any pair, i and j, from P' (randomly, irrespective of fronts)

   b) Identify the better of these two chromosomes. Chromosome i is better than chromosome j if:

   $I_{i,\,rank} \neq I_{j,rank}$: $I_{i,\,rank} < I_{j,rank}$

43

$$I_{i, \text{rank}} = I_{j,\text{rank}} \colon I_{i,\text{dist}} > I_{j,\text{dist}}$$

    c) Copy (without removing from P') the better of these two chromosomes in a new box, P''

    d) Repeat till P'' has N_p members

    Not all of P' need be in P''. By this method, the better members of P' are copied into P'' stochastically

6. Copy all of P'' in a new box, D, of size N_p.

    Carry out crossover (using the stochastic remainder roulette-wheel selection procedure)[3] and mutation[3] of chromosomes in D.

    This gives a box of N_p *daughter chromosomes*.

7. Map adaptively (as in Step 2) and evaluate fitness values (with penalty) for all daughter chromosomes.

8. **Elitism:** Copy all the N_p best parents (P'') and all the N_p daughters with transposons (D) into box PD. Box PD has $2N_p$ chromosomes

    a) Reclassify these $2N_p$ chromosomes into fronts (box PD') using *only* non-domination (as described in Step 2 above)

    b) Take the *best* N_p from box PD' and put into box P'''

9. This completes one generation. Stop if appropriate criteria are met, e.g., the generation number > maximum number of generations (user specified).

10. Copy P''' into starting box, P. Go to Step 2 above.

- - - - - -

Details of NSGA-II are from Deb[4] and Deb et al.[19]

```
                    ┌─────────────────────┐
                    │   N_gen = 0         │
                    └─────────────────────┘
                              │
                              ▼
              ┌──────────────────────────────────┐
              │ Box P (N_p): Generate N_p parents,│
              │ discretize in specified range, map│
              │ adaptively, curve-fit to obtain   │
              │ continuous functions and solve model│
              │ equations.                        │
              └──────────────────────────────────┘
                              │
                              ▼
              ┌──────────────────────────────────┐
              │ Box P' (N_p): Classify and calculate│
              │ I_rank and I_dist of chromosomes in P.│
              │ Apply penalty whenever required.  │
              └──────────────────────────────────┘
                              │
                              ▼
              ┌──────────────────────────────────┐
              │ Box P'' (N_p): Copy best N_p from │
              │ P'                                │
              └──────────────────────────────────┘
                              │
                              ▼
              ┌──────────────────────────────────┐
              │ Box D (N_p): Do crossover and     │
              │ mutation of chromosomes in P''    │
              └──────────────────────────────────┘
                              │
                              ▼
              ┌──────────────────────────────────┐
              │ Box PD (2 N_p): Combine P'' and D │ ·······┐
              └──────────────────────────────────┘        :
                              │                            :
                              ▼                            :
              ┌──────────────────────────────────┐        :
              │ Box PD' (2 N_p): Put PD into fronts│       : Elitism
              └──────────────────────────────────┘        :
                              │                            :
                              ▼                            :
              ┌──────────────────────────────────┐        :
              │ Box P''' (N_p): Select best N_p from│·······┘
              │ PD' .                             │
              └──────────────────────────────────┘
                              │
                              ▼
              ┌──────────────────────────────────┐
              │ N_gen = N_gen + 1                 │
              │ P''' → P                          │
              └──────────────────────────────────┘
```
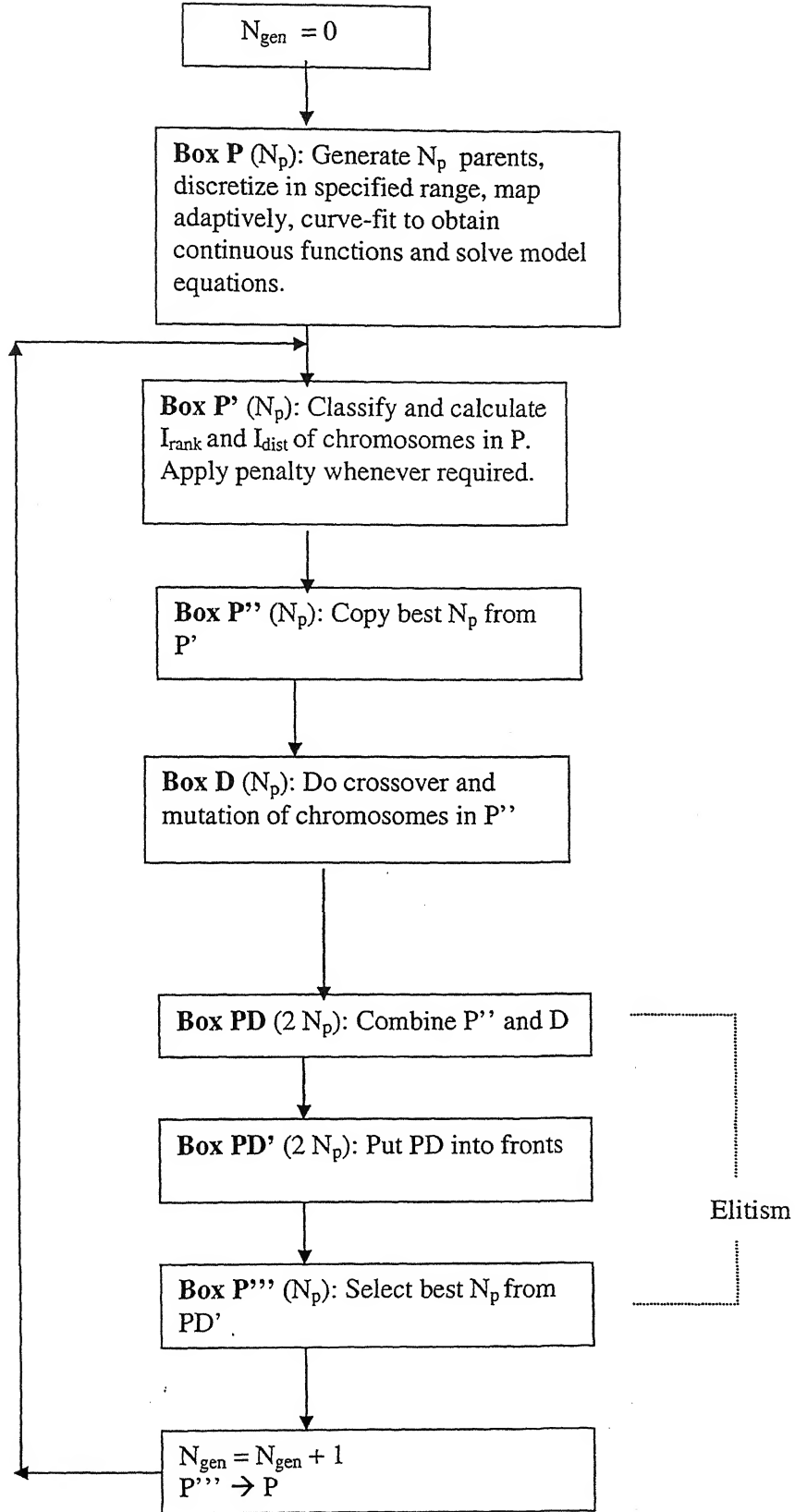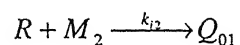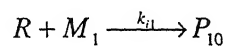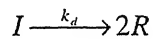
Figure A1.1: Flowchart of elitist NSGA-II adapted for continuous variables

45

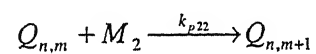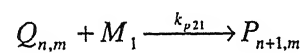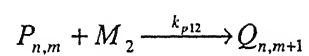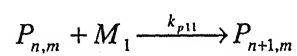# Appendix 2

Kinetic Scheme

Initiation:

$$I \xrightarrow{k_d} 2R$$

$$R + M_1 \xrightarrow{k_{i1}} P_{10}$$

$$R + M_2 \xrightarrow{k_{i2}} Q_{01}$$

Propagation:

$$P_{n,m} + M_1 \xrightarrow{k_{p11}} P_{n+1,m}$$

$$P_{n,m} + M_2 \xrightarrow{k_{p12}} Q_{n,m+1}$$

$$Q_{n,m} + M_1 \xrightarrow{k_{p21}} P_{n+1,m}$$

$$Q_{n,m} + M_2 \xrightarrow{k_{p22}} Q_{n,m+1}$$

Termination by Combination:

$$P_{n,m} + P_{r,q} \xrightarrow{k_{tc11}} D_{n+r,m+q}$$

$$P_{n,m} + Q_{r,q} \xrightarrow{k_{tc12}} D_{n+r,m+q}$$

$$Q_{n,m} + Q_{r,q} \xrightarrow{k_{tc22}} D_{n+r,m+q}$$
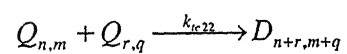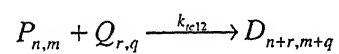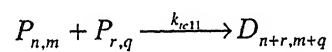
Termination by Disproportionation:

$$P_{n,m} + P_{r,q} \xrightarrow{k_{td11}} D_{n,m} + D_{r,q}$$

$$P_{n,m} + Q_{r,q} \xrightarrow{k_{td12}} D_{n,m} + D_{r,q}$$

$$Q_{n,m} + Q_{r,q} \xrightarrow{k_{td22}} D_{n,m} + D_{r,q}$$

46
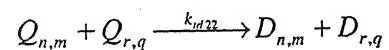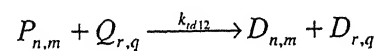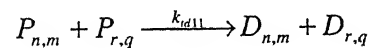
(Appendix 2...Contd....b)

Chain Transfer:

$$P_{n,m} + M_1 \xrightarrow{\;k_{f11}\;} D_{n,m} + P_{10}$$

$$P_{n,m} + M_2 \xrightarrow{\;k_{f12}\;} D_{n,m} + Q_{01}$$

$$Q_{n,m} + M_1 \xrightarrow{\;k_{f21}\;} D_{n,m} + P_{10}$$

$$Q_{n,m} + M_2 \xrightarrow{\;k_{f22}\;} D_{n,m} + Q_{01}$$

* * *
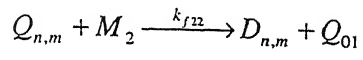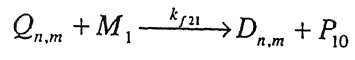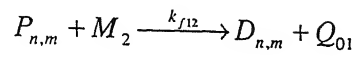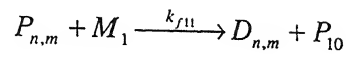
Model Equations[12]

Reactor Mass Balances:

$$\frac{dM_1}{dt} = \frac{u}{V}(M_{1f} - M_1) - [(k_{p11} + k_{f11})P_0 + (k_{p21} + k_{f21})Q_0]M_1 \tag{A1}$$

$$\frac{dM_2}{dt} = \frac{u}{V}(M_{2f} - M_2) - [(k_{p22} + k_{f22})Q_0 + (k_{p12} + k_{f12})P_0]M_2 \tag{A2}$$

$$\frac{dI}{dt} = \frac{u}{V}(I_f - I) - k_d I \tag{A3}$$

$$\frac{dV}{dt} = u \tag{A4}$$

Live Polymer Moments:

$$P_0 = \left[\frac{2fk_d I}{(k_{tc11} + k_{td11}) + 2\beta(k_{tc12} + k_{td12}) + \beta^2(k_{tc22} + k_{td22})}\right]^{1/2} \tag{A5}$$

$$\text{where} \quad \beta = \frac{k_{p12} + k_{f12}}{(k_{p21} + k_{f21})\phi} \quad ; \quad \phi = \frac{M_1}{M_2}$$

$$Q_0 = \beta P_0 \tag{A6}$$

$$P_1 = \left[\frac{w_1 C_1 \alpha_1 + \frac{\alpha_1 \gamma}{r_1} Q_1 + w_1(\alpha_1 P_0 + \frac{\alpha_1 \gamma}{r_1} Q_0)}{(1 - \alpha_1)}\right] \tag{A7}$$

$$Q_1 = \left[\frac{w_2 C_2 \alpha_2 + \frac{\alpha_2}{\gamma r_2} P_1 + w_2(\alpha_2 Q_0 + \frac{\alpha_2}{\gamma r_2} P_0)}{(1 - \alpha_2)}\right] \tag{A8}$$

(Appendix 3...contd....b)

$$P_2 = \left[\frac{w_1{}^2 C_1 \alpha_1 + \dfrac{\alpha_1 \gamma}{r_1} Q_2 + 2 w_1 \alpha_1 P_1 + \dfrac{2 w_1 \alpha_1 \gamma}{r_1} Q_1 + w_1{}^2 (\alpha_1 P_0 + \dfrac{\alpha_1 \gamma}{r_1} Q_0)}{(1 - \alpha_1)} \right] \tag{A9}$$

$$Q_2 = \left[\frac{w_2{}^2 C_2 \alpha_2 + \dfrac{\alpha_2}{\gamma r_2} P_2 + 2 w_2 \alpha_2 Q_1 + \dfrac{2 w_2 \alpha_2}{\gamma r_2} P_1 + w_2{}^2 (\alpha_2 Q_0 + \dfrac{\alpha_2}{\gamma r_2} P_0)}{(1 - \alpha_2)} \right] \tag{A10}$$

where

$$C_1 = \frac{(k_{f11} P_0 + k_{f21} Q_0)}{k_{p11}} \qquad ; \qquad C_2 = \frac{(k_{f12} P_0 + k_{f22} Q_0)}{k_{p11}}$$

$$r_1 = \frac{k_{p11}}{k_{p12}} \qquad ; \qquad r_2 = \frac{k_{p22}}{k_{p21}} \qquad ; \qquad \gamma = \frac{k_{p21}}{k_{p12}}$$

$$\alpha_1 = \left\{ \frac{k_{p11} M_1}{[(k_{p11} + k_{f11}) M_1 + (k_{p12} + k_{f12}) M_2 + (k_{tc11} + k_{td11}) P_0 + (k_{tc12} + k_{td12}) Q_0]} \right\}$$

$$\alpha_2 = \left\{ \frac{k_{p22} M_2}{[(k_{p22} + k_{f22}) M_2 + (k_{p21} + k_{f21}) M_1 + (k_{tc22} + k_{td22}) Q_0 + (k_{tc12} + k_{td12}) P_0]} \right\}$$

Dead Polymer Moments:

$$\frac{d\lambda_0}{dt} = (\frac{1}{2} k_{tc11} + k_{td11}) P_0{}^2 + (\frac{1}{2} k_{tc22} + k_{td22}) Q_0{}^2 + (k_{tc12} + 2 k_{td12}) P_0 Q_0$$

$$+ (k_{f11} M_1 + k_{f12} M_2) P_0 + (k_{f22} M_2 + k_{f21} M_1) Q_0 - \frac{\lambda_0}{V} u \tag{A11}$$

$$\frac{d\lambda_1}{dt} = (k_{tc11} P_0 + k_{td11} P_0 + k_{tc12} Q_0 + k_{td12} Q_0 + k_{f11} M_1 + k_{f12} M_2) P_1$$

$$+ (k_{tc22} Q_0 + k_{td22} Q_0 + k_{tc12} P_0 + k_{td12} P_0 + k_{f22} M_2 + k_{f21} M_1) Q_1 - \frac{\lambda_1}{V} u \tag{A12}$$

(Appendix 3…contd….c)

$$\frac{d\lambda_2}{dt} = (k_{tc11}P_0 + k_{td11}P_0 + k_{tc12}Q_0 + k_{td12}Q_0 + k_{f11}M_1 + k_{f12}M_2)P_2$$

$$+ (k_{tc22}Q_0 + k_{td22}Q_0 + k_{tc12}P_0 + k_{td12}P_0 + k_{f22}M_2 + k_{f21}M_1)Q_2$$

$$+ k_{tc11}P_1^2 + k_{tc22}Q_1^2 + 2k_{tc12}P_1Q_1 - \frac{\lambda_2}{V}u \qquad (A13)$$

Properties, etc.:

$$F_1 = \frac{r_1\phi^2 + \phi}{r_1\phi^2 + 2\phi + r_2} \qquad (A14)$$

$$x_1 = \frac{V_0M_{10} + \int_0^t u(t)M_{1f}dt - VM_1(t)}{V_0M_{10} + \int_0^t u(t)M_{1f}dt} \qquad (A15)$$

$$M_n = \frac{\lambda_1}{\lambda_0} \qquad (A16)$$

$$M_w = \frac{\lambda_2}{\lambda_1} \qquad (A17)$$

$$Q = \frac{M_w}{M_n} \qquad (A18)$$

$$* * *$$

50

Appendix 4

Program Listing

The listings of computer programs are available with the author and Prof. S. K. Gupta, Department of Chemical engineering, Indian Institute of Technology, Kanpur, India.